Ch. 6: Solute transport

Table of Contents

6.1 Introduction 2

6.2 Definition of a chemical 2

6.3 Transport in the matrix soil (primary domain) 4

6.3.1 The advection-dispersion equation in 1D 4

6.3.2 The advection-dispersion Equation in 2D 6

6.3.3 Alternative solute transport models 7

6.4 Boundary conditions and inputs 7

6.4.1 Top boundary 7

6.4.2 Lower boundary – leaching to groundwater 8

6.4.3 Initial conditions 8

6.4.4 Release in the soil 9

6.5 Leaching to drains 9

6.6 Root uptake 9

6.7 Sorption 10

6.7.1 Equilibrium isotherms 11

6.7.2 Instantaneous sorption 12

6.7.3 Time dependent sorption 12

6.7.4 Multiple levels of sorption 13

6.8 Transport in primary and secondary domain 14

6.8.1 Water content and flow in two domains 15

6.8.2 Solute concentration and flow in two domains 17

6.8.3 Initial concentrations in two domains 17

6.8.4 The sink terms in two domains 18

6.8.5 Sorption in two domains 18

6.9 Transport in biopores 19

6.10 Colloids 21

6.10.1 Generation of colloids 21

6.10.2 Adsorption to colloids 26

6.10.3 Transport of colloids 26

6.10.4 Filtration of colloids 27

6.11 Numerical solutions 27

6.12 Parameter overview 28

6.1 References 42

6.1 Introduction

The objective of the solute balance model is to keep track of solutes allocated to the system or released or generated in the system. The allocation and the fate of solutes above the soil surface are described in Chapter 3. The soil solute balance, described here, keeps track of how solutes are distributed within a soil profile. The model considers sorbed solutes as well as solutes in the soil solution. In the soil, transport, decomposition, sorption, diffusion, and crop uptake are major processes affecting solute fate. The decomposition can be influenced by temperature, water content, concentration of the compound, CO_2 concentration and depth. Additionally, decomposition can have a lag-phase. A compound can break down into metabolites.

Ammonium, nitrate, tracers, pesticides and plant toxins are specialisations of the solute balance. This section focuses on the common solute transport and transformation in the soil system, the special case of ammonium and nitrate are described in Chapter 7 and the special case of pesticides and plant toxins are described in Chapter 8.

6.2 Definition of a chemical

To function as a solute in Daisy, a chemical must be defined. Some examples are available under “lib” in the Daisy directory (see *chemistry-base.dai* and *chemistry.dai*). Depending on the chemical and the relevant processes different parameters can be defined. For the parameters affecting above-ground processes, see Chapter 3.

The chemicals are defined in a hierarchical system, where the sub-groups inherit the parameter definitions from the “parent”-definition (Figure 6.1). The highest level definition is named *base*. Three lower level “parent” definitions based on *base* exist: *FOCUS, default,* and *nutrient. FOCUS* and *nutrient* are described in Chapter 8 and 7, respectively. Under *default* four chemicals are parameterized: *classic,* which can be used to represent a chemical where no specific information is available, *solid* which represents non-dissolvable chemicals, *colloid* which represent soil colloids, and *tracer* which represent a chemical with no sorption, plant uptake or degradation.

User specified new chemicals can be parameterized based on any of the existing parameterizations and will then inherit the parameters of the “parent”-chemical.

The fate (transport, adsorption, transformation, etc.) is traced and, if relevant, combined with chemical reactions in the *chemistry* component. Thus, the *default chemistry* model takes two entries: *trace* and *reaction*. These are defined with a list of chemicals that should be traced (e.g. calculated and logged) and a list of chemical reactions that should be simulated, respectively.

Like the chemical parameterizations, there are a number of predefined *chemistry* models where the reactions relevant to the traced chemical are defined as default (e.g. the chemistry models *DOM, N* and *pesticides)*. Additionally, the *chemistry multi* model makes it possible to combine the simulation of multiple chemicals and chemical reactions, e.g. both N and pesticides or multiple pesticides. The *multi* model also takes several parameters that control the amount of chemical that can be removed within a timestep (*max_sink_total, max_sink_solute, max_sink_secondary, min_sink_total,* see more in parameter overview). Thus, if variable timesteps is enabled[1], the timestep will be scaled down in order to ensure that no more chemical than defined by the parameters are removed by drains or biopores within a timestep. The *nutrient* model is a variation of the *multi* model where the N chemistry automatically is included. The *nutrient* model is the default *chemistry* model*;* thus, N dynamics are simulated as standard (see more in Chapter 7). If it is wished to run the simulation without accounting for N dynamics *chemistry* should be defined with the *none* model.

Parameterisations and processes specific to plant toxins and pesticides are described in Chapter 8. A pesticide can, to the largest extent possible, follow the parameterisation requirements used in FOCUS pesticide guidelines (FOCUS, 2012), see Chapter 8 and Appendix 3.3.

Parameterisations and processes specific for mineral N are described in Chapter 7.

Parameters and processes specific for C and DOC are described in Chapter 9.

The most important parameters for solute fate in the soil are decomposition, sorption, diffusion coefficient, and the crop uptake reflection factor. The different parameters are described under the relevant process description and a complete list of parameters can be found in the parameter table at the end of the chapter and in the reference manual.

The solute transport description in this chapter is for a non-sorbing, non-decaying chemical, like a tracer. In section 6.7 chemical sorption is described, whereas chemical transformations are described separately for N and pesticides in Chapter 7 and 8, respectively.

---

[1] Variable timestep can be enabled by setting the *minimal_timestep* (at top level, under the Daisy program component) to smaller timestep than the *timestep* parameter. By default *minimal_timestep* is equal to *timestep* and *timestep* is 1 hour.

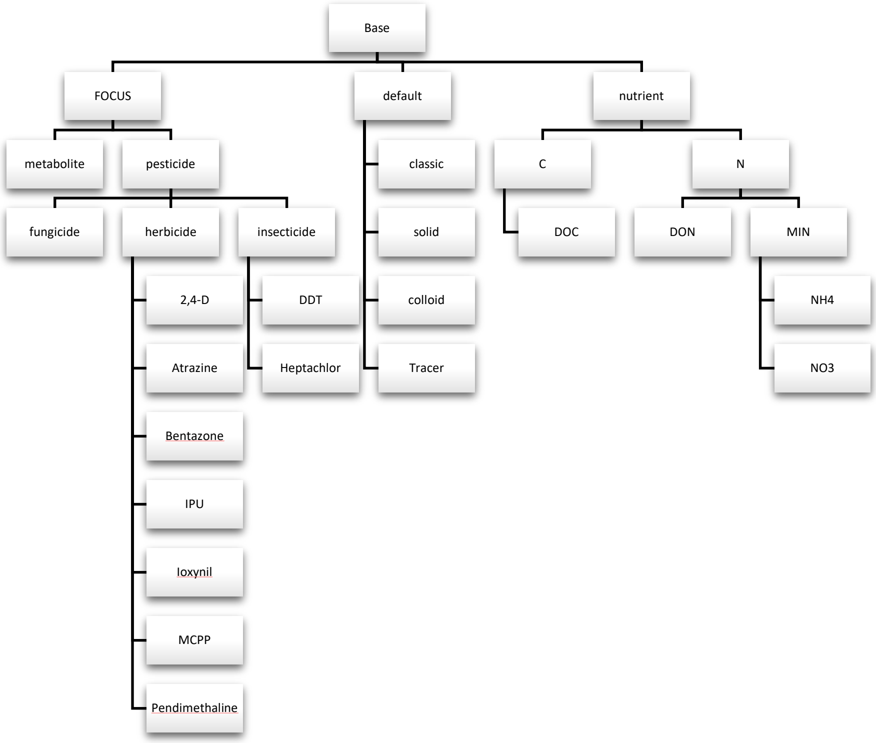


Figure 6.1: The hierarchical organization of the defined chemicals in Daisy.

6.3 Transport in the matrix soil (primary domain)

6.3.1 The advection-dispersion equation in 1D

The core of the soil solute balance model is the convection-dispersion equation, here shown for a non-sorbing tracer:

$$\frac{\partial(\theta C)}{\partial t} = \frac{\partial}{\partial z}\left(\theta D \frac{\partial C}{\partial z} - qC\right) + \Sigma\Gamma \tag{6.1}$$

where:

C = the solute concentration in the liquid phase [g cm^{-3}],

Z = the vertical spatial coordinate [cm] (considered positive upwards),

D = the diffusion-dispersion coefficient [cm^2 h^{-1}],

q = the Darcy water flux [cm^2 h^{-1}] and

$\Sigma\Gamma$ = the sum of the sink-source-terms [g cm^{-3} h^{-1}]. The sink-source-terms are positive for inputs and negative for outputs and include:

- Root uptake, Γ_{RWU}^{S}, described in section 6.6.
- Transport to drain, Γ_{Df}^{S}, described in section 6.5.
- Transport to biopores, Γ_{Bf}^{S}, described in section 6.9.
- Exchange between domains within matrix water flow, $\Gamma_{d1\rightarrow d2}^{S}$, described in section 6.8.

For chemicals subject to sorption, decomposition and transformation, chemical the sink-source-terms can also include:

- Sorption and desorption, Γ_{At}^{S}, described in section 6.7.
- Decomposition, Γ_{D}^{S}, described in Chapters 7 and 8.
- Transformation, Γ_{met}^{S}, e.g. creation of metabolites, described in Chapter 8.
- Exchange with colloids, Γ_{col}^{S}, described in section 6.10.

The diffusion-dispersion coefficient

The diffusion dispersion coefficient includes the effects of both the hydrodynamic dispersion and diffusion calculated as:

$$D = \lambda \frac{q}{\theta} + \tau D_0 \tag{6.2}$$

where:

λ = the dispersity or dispersion length [cm], by default 5 [cm]. Note that the discretisation used should not be larger than $2 \cdot \lambda$ to avoid dispersion due to the grid system,

τ = the tortuosity factor [-] and

D_0 = the diffusion coefficient of the solute in free solution [cm^2 h^{-1}].

The tortuosity factor

The tortuosity factor, τ, is by default calculated as a linear impedance factor (Nye and Tinker, 1977):

$$\tau = a + b\theta \tag{6.3}$$

where:

θ = the volumetric soil water content [cm^3 cm^{-3}]

a = θ offset [cm^3 cm^{-3}]. a corresponds, by default, to the wilting point, and

b = θ factor [-]. Default = 2.

Alternatively, the tortuosity factor can be calculated using the approach by (Millington and Quirk, 1961):

$$\tau = \frac{\theta^{7/3}}{\theta_s} \tag{6.4}$$

where θ_s = the saturated water content [cm^3 cm^{-3}].

The 1D transport vector

Vertical movement of a solute in the soil matrix flow regime is a result of convection as well as diffusion. Thus, the flux density of the solute, e.g. the transport vector in 1-dimension, J [g cm^{-2} h^{-1}] is calculated as:

$$J = -\theta D \frac{\partial C}{\partial z} + qC \tag{6.5}$$

The 1D convection-dispersion equation is implemented as the solute transport model termed *Hansen* under *movement – vertical – matrix solute.* The *matrix solute* component takes a sequence of solute transport models. For 1D this is: *Hansen, convection, none.* The last two are described below. If the convection-dispersion equation (The *Hansen* model) does not converge within a specified number of iterations the more simplified model *convection* is used. Similarly, if a solution cannot be found only by applying *convection,* the *none* model is used. If no solution can be found, the simulation terminates.

6.3.2 The advection-dispersion Equation in 2D

In 2-dimensions, both vertical and horizontal solute transport due to advection and dispersion is considered. As for water movement in 2-dimensions (see Chapter 4) the x-axis is chosen in the horizontal direction and z is chosen in the vertical direction, positive upwards. Thus, the advection-dispersion equation for a non-sorbing tracer in 2D is written as

$$\frac{\partial(\theta C)}{\partial t} = -\nabla \cdot (C\boldsymbol{q} - \theta \boldsymbol{D} \nabla C) + \Sigma\Gamma \tag{6.6}$$

where:

$\boldsymbol{q}$ = the water flux density matrix and

$\boldsymbol{D}$ = the dispersion matrix expressed as:

$$\boldsymbol{D} = \begin{bmatrix} D_{xx} & D_{xz} \\ D_{zx} & D_{zz} \end{bmatrix} \tag{6.7}$$

The dispersion matrix

The elements in $\boldsymbol{D}$ are calculated as (Mollerup et al., 2014):

$$\begin{gathered} D_{xx} = \alpha_L \frac{v_x^2}{|\boldsymbol{v}|} + \alpha_T \frac{v_z^2}{|\boldsymbol{v}|} + D^* \\ D_{zz} = \alpha_L \frac{v_z^2}{|\boldsymbol{v}|} + \alpha_T \frac{v_x^2}{|\boldsymbol{v}|} + D^* \\ D_{xz} = D_{zx} = (\alpha_L - a_T)\frac{v_x v_z}{|\boldsymbol{v}|} \end{gathered} \tag{6.8}$$

where:

$\boldsymbol{v} = \boldsymbol{q}/\theta$ = the mean velocity in the pores [cm h^{-1}],

v_x and v_z = the velocity component in the x and z direction [cm h^{-1}], respectively,

α_L and α_T = the longitudinal and transversal dispersion [cm h^{-1}], respectively, and

D^* = the molecular diffusion [cm h^{-1}], which can be expressed as $D^* = \tau D_0$ where D_0 is the diffusion coefficient and τ is the tortuosity factor, calculated with Eq. 6.3 or 6.4.

The 2D transport vector

When the solution to the advection-dispersion equation is found, the solute flux can be determined by:

$$\boldsymbol{J} = C\boldsymbol{q} - \theta \boldsymbol{D} \nabla C \tag{6.9}$$

where $\boldsymbol{J}$ [g cm^{-2} h^{-1}] is the 2D transport vector.

The 2D advection dispersion equation is implemented as the solute transport model termed *Mollerup* under *movement – rectangle – matrix solute.* The *matrix solute* component takes a sequence of models which in 2D are *Mollerup, convection, none.* As for 1D the models will be tried in turn and if none converge the simulations will terminate.

The *Mollerup* model takes the parameters: *solver, debug, stabilizing_method, enable_boundary_conditions,* and *upstream_weight,* all described in section 6.11 and in the parameter overview.

6.3.3 Alternative solute transport models

The convection transport model

The *convection* model only accounts for solute transport by convection. Thus, transport is calculated as pure forward transport following the water flow calculated by Richards Equation (see Chapter 4):

$$\boldsymbol{J_{edge}} = q_{edge} * C_{old} \tag{6.10}$$

where:

$\boldsymbol{J_{edge}}$ = the solute transport vector through the upper edge of the node.

q_{edge} = the water flow through the upper edge of the node and

C_{old} = the solute concentration in the node above.

This option is applied for all nodes and transport through the lower boundary, but transport to the upper boundary (the soil surface) is calculated as normal (section 6.4.1). The convection transport model is only applied if the advection-dispersion model fails to find a solution.

The none transport model

In the *none* model all solute transport is disabled, except the transport through boundaries which is calculated as normal (section 6.4). The *none* model is only applied if both the convection and the advection-dispersion models fails.

6.4 Boundary conditions and inputs

The characteristics of the boundaries of the simulated domain in Daisy vary depending on their position.

Side boundaries: no flow

The side-boundaries are defined as no-flow boundaries and there is no exchange of solute over the side-boundaries, whereas the top boundary is defined by the surface processes (see Chapter 3) and the exchange over the bottom boundary is defined with either free drainage, a fixed or fluctuating groundwater table or pressure, an aquitard, or a lysimeter boundary. A solute concentration for the groundwater below the soil column can be specified.

6.4.1 Top boundary

Solutes added with overhead- or surface irrigation or spread directly on the surface will end up on the soil surface and be introduced to the soil from there. A mixing layer [default 0.1 cm] constitutes the top boundary between the above-ground and the soil (see Chapter 3).

Infiltration

The mass of solute at the surface available for infiltration to the soil is the mass of solute in ponded water, if present, plus the dissolved mass stored in the mixing layer. Thus, the infiltrating mass to the soil, $J^{C}_{w,Soil\_inf}$ [g m^{-2} h^{-1}], is calculated as a function of the dissolved solute concentration in the surface water (ponded and in mixing layer) and the amount of infiltrating water as:

$$J^{C}_{w,Soil\_inf} = q_{pond\_inf} \cdot C_{pond} \tag{6.11}$$

Where:

q_{pond_inf} = infiltrating water (see Chapter 4) [cm h^{-1}] and

C_{pond} = concentration of solute in surface water [g m^{-2} cm^{-1}]

Exfiltration

Solutes in the soil can exfiltrate to the surface water (ponded and in mixing layer) if the concentration in the soil water is higher than the concentration in the surface water. The exfiltration is calculated based on the difference in concentration on the surface (storage concentration) and in the topsoil (default the top 0.1 cm). The exfiltration can be delayed, using a resistance factor *Rmix* [default $1{\cdot}10^{9}$, [h mm^{-1}], so that the maximum amount that can exfiltrate to the surface water is *(storage concentration- soil concentration)/Rmix*.

Incorporation

Additionally, chemicals on the surface (in a litter layer, in ponded water and in the mixing layer) can be incorporated in the soil with the management operations ploughing and harrowing, where topsoil is mixed or swapped with deeper soil layers (see Chapter 11).

Chemicals, except N, taken up by the plant will not be released from the plant again even though plant residues are left on the field and incorporated in the soil. Thus, plant uptake of chemicals is treated as a sink term where the chemical taken up by the plant is considered to leave the system.

6.4.2 Lower boundary

The lower boundary is by default defined with the same concentration as in the lowest cell of the soil column (e.g. zero gradient between soil column and groundwater). Thus, transport of chemicals over the lower boundary will only be calculated by advection as a function of water flow in and out of the soil column.

Alternatively, a concentration of solutes below the soil column can be specified with the parameter *C_below* and exchange of solute over the lower boundary will be calculated by both advection and diffusion.

6.4.3 Initial conditions

For most solutes it is assumed that they are released to or generated in the system during the simulation. Thus, the *initial* [g cm^{-3}] parameter is by default 0. However, an initial concentration can be specified with the parameter *initial_C* (sequence) or *initial_C_plf* .This is, for example, the case for N, where an assumption of a zero initial concentration would be erroneous, see Chapter 7. Another example would be if there was knowledge of, for example, pesticide residues in the soil profile.

Initial concentration

Both *initial_C* and *initial_C_plf* take a sequence of value pairs describing soil depth and concentration. *initial_C* reads the sequence from the top going down and *initial_C_plf* from the bottom going up. Thus, *initial_C* takes a sequence of (END VALUE) pairs, where END [cm] is the end of the relevant soil layer (a negative number) and VALUE [g cm^{-3}] is the solute concentration of water in that soil layer; and *initial_C_plf* takes a sequence of (DEPTH VALUE) pairs, starting from the bottom and going up. Alternatively, the initial concentration can be given by the optional state variable *C* [g cm^{-3}]. Initial concentrations may also be specified separately for different flow domains, see section 6.8.

Initial mass

The amount of chemical in the soil column can also be defined as the total mass of the chemical per volume water, soil and air [g cm^{-3}]. Then the initial amount can be specified with the parameters *initial_M* (sequence) and *initial_M_plf.* They, as the parameters for concentration, take a sequence of value pairs describing soil depth and mass going either from the top down (*initial_M)* or from the bottom up (*initial_M_plf).* Alternatively, the initial mass can be given by the optional state variable *M* [g cm^{-3}]. Initial masses may also be specified separately for different flow domains, see section 6.8.

Remove solutes

Conversely, the *remove_solute* action can be applied if it is wished to ensure that no chemical is left in the field. For example, if simulating recurring applications that should not interfere. The remove solute's action takes one parameter (*chemical*) which is the name of the chemical to be removed.

6.4.4 Release in the soil

Chemicals can enter the soil directly with subsoil irrigation or if they are created by transformation from another chemical. The latter process is described in chapter 8.

Subsoil irrigation can be parameterized using the *irrigation_subsoil* action, by specifying the chemical and amount [ppm] under *solute* (see chapter 11).

Alternatively, can the chemical state variable *S_permanent* [g cm^{-3} h^{-1}] be defined directly, specifying a constant input in the soil of a given chemical.

6.5 Leaching to drains

Solute transport to drains, $\Gamma_{s,drain}$ [g cm^{-1} h^{-1}], is both in 1-Dimension (Hooghoudt's equation) and 2-Dimensions (Dynamic drainage model) (see Chapter 4) calculated by simple advection as:

$$\Gamma_{drain}^{s} = \Gamma_{Df} \cdot C \qquad (6.12)$$

where:

Γ_{Df} = drain water flow [cm h^{-1}] and

C = solute concentration in the water [g cm^{-3}].

6.6 Root uptake

Solutes may be taken up by plants by as an active or a passive process. If the process is active, it is assumed that the plant regulates the concentration at the root surface of the considered chemical, and that both mass flow and diffusion may contribute to the movement of solutes to the root surface. For now (Daisy 7) only N-uptake is simulated as an active process and active root uptake is therefore described in Chapter 7.7. However, in the future additional nutrients could be included in Daisy and plant uptake of these could then also be simulated with an active process. During passive root uptake, the movement of solutes to the root surface is simulated as pure mass flow, e.g. the solute is carried with the transpiration stream to the root surface where it is taken up. The passive uptake process is used to account for possible plant uptake of traces, pesticides, and plant toxins.

The transport of solutes from the bulk soil to the root surface is based on several assumptions like those for water flow (Chapter 4). Each root may exploit an average effective volume of soil which is assumed to be a cylinder around the root. The radius of this cylinder is assumed to correspond to the average half distance between the roots. Thus, the passive solute uptake is a result of the water flow towards the root and the solute concentration in this cylinder. However, solutes may not always enter the root when arriving at the root surface. This is described by introducing a reflection coefficient when calculating the root solute uptake, hence solute uptake is calculated as:

$$I_s = (1 - rc) q_r C \tag{6.13}$$

where:

I_s = the solute uptake per unit root length [g cm^{-2} h^{-1}]

q_r = the water flux towards the root surface [cm h^{-1}] (see chapter 4)

C = solute concentration in the water [g cm^{-3}]

rc = the reflection coefficient, going from no reflection, e.g. full uptake at 0 to full reflection, e.g. no uptake at 1 (default value rc = 1).

The solute uptake is summarized over the entire root length to calculate the sink-term in the advection-dispersion equation (Eq. 6.1).

6.7 Sorption

Chemicals in the soil can sorb to organic matter and clay particles. The equilibrium between chemicals in solution and chemicals sorbed to organic matter and clay particles can be simulated by applying a Freundlich, Langmuir, or Linear isotherm. Additionally, full and no adsorption can be simulated.

Sorption can both be simulated as instantaneous (in equilibrium within one time step) or as time dependent. Time dependent adsorption is simulated as a reaction between two parameterizations of the same chemicals.

Full adsorption

Full adsorption is used to simulate non-solutes that are fully adsorbed to the soil. This is the default adsorption model for non-adsorbable chemicals parameterized as *solid* (section 6.2).

No adsorption

The adsorption model *none* is used to simulate solutes that do not adsorb to the soil, for example, tracers.

Both the *full* and *none* sorption models can be relevant to use when simulating kinetic time dependent sorption (section 6.7.3). In that case, they are not considered as sorption models, but rather as two forms of the same chemical between which reactions may occur.

6.7.1 Equilibrium isotherms

In Daisy three equilibrium isotherms are implemented to describe sorption equilibrium: *Freundlich, Langmuir,* and *Linear*.

Freundlich adsorption

Adsorption according to the Freundlich isotherm is calculated as (FOCUS, 2012; Freundlich and Heller, 1939):

$$A_{eq} = \rho * K_{fd} * C_{ref} \left(\frac{C}{C_{ref}} \right)^{m} \tag{6.14}$$

where:

A_{eq} = the adsorbed chemical [g g^{-1}] at equilibrium.

ρ = dry bulk density [g cm^{-3}].

K_{d_f} = soil adsorption coefficient [cm^3 g^{-1}].

C_{ref} = reference concentration [g cm^3], used to determine the other values. According to FOCUS (FOCUS, 2012) the standard values is 1 mg L^{-1} = $1e^{-6}$ = 1 ppm.

m = the freundlich exponent [-].

K_{fd} is calculated from the organic carbon/water distribution coefficient, K_{fOC} [cm^3 g], and the clay/water distribution coefficient, K_{fclay} [cm^3 g] as:

$$K_{fd} = Cl * K_{fclay} + OM * OM_c * K_{fOC} \tag{6.15}$$

where:

Cl = the clay fraction (< 2 µm) [fraction]

OM = organic matter fraction [fraction]

OM_c = the carbon fraction of organic matter = 0.587 [fraction]

If only K_{fOC} is specified, K_{fclay} defaults to 0 as it is assumed that K_{fOC}, when measured, accounts for sorption to both organic carbon and clay particles or that $K_{fOC} >> K_{fclay}$ and K_{fclay} can thus be neglected. If only K_{fclay} is specified $K_{fOC} = K_{fclay}$ as the best guess for K_{fOC}.

Langmuir adsorption

Adsorption following the Langmuir isotherm is calculated as:

$$A_{eq} = \rho * \frac{a_{max} * C}{1/K_l + C} \tag{6.16}$$

where

K_l = the half saturation constant [cm^{-3}/g] and

a_{max} = the soil's max adsorption capacity [g/g]. This is calculated from the max adsorption capacity to clay, a_{max_clay}, and organic carbon, a_{max_OC}, as done for the Freundlich soil adsorption coefficient, K_{fd} (Eq. 6.14).

Linear adsorption

Adsorption following a linear isotherm is calculated as:

$$A_{eq} = \rho * K_{ld} * C \tag{6.17}$$

where:

K_{ld} = the soil dependent distribution parameter [cm^3 g^{-1}]

K_{ld} can either be defined directly as *K_d* or it can be calculated from the clay dependent distribution parameter and the organic carbon distribution parameter as done for the Freundlich soil adsorption coefficient, K_{fd} (Eq. 6.14).

6.7.2 Instantaneous sorption

When it is assumed that equilibrium between sorbed and dissolved chemical is obtained within a timestep, adsorption is calculated as instantaneous sorption according to one of the equilibrium isotherms or as full adsorption. Thus, the adsorbed amount, A [g g^{-1}], is always = A_{eq}, and the dissolved amount is given by

$$\theta * C = M - S \tag{6.18}$$

where:

M = the total mass of chemical [g]

For instantaneous sorption the equilibrium between adsorbed, A_0, and dissolved chemical, C_0, is calculated between timesteps. Then, the advection-dispersion equation (see section 6.3) is solved for the dissolved chemical, resulting in a new concentration, C_1*. Based on C_1* the equilibrium between the adsorbed, A_1 and dissolved, C_1, chemical is calculated and the advection-dispersion equation is then solved for C_1*, resulting in a new concentration, C_2*, and so on. Thus, the calculations of adsorption and transport are decoupled. Physically it would be more correct to calculate adsorption as part of the advection-dispersion equation. However, the decoupled approach makes it possible to easily change between different adsorption models.

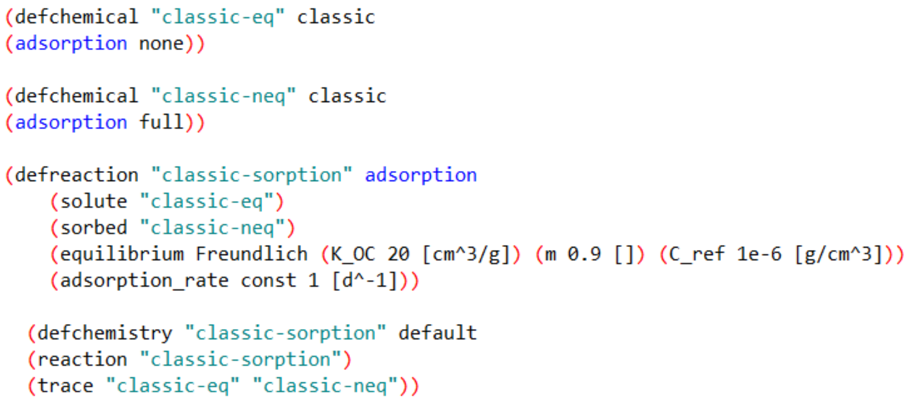
6.7.3 Time dependent sorption

Sorption kinetics can be simulated as a reaction between two forms of the same chemical. Thus, the two parameterizations of the chemical represent the same chemical and they only differ in terms of sorption (see example below). Here, two chemicals (based on the *classic* definition in *chemistry-base.dai*) are parameterized, one (*classic-eq*) fully dissolved (*adsorption none*) and one (*classic-neq*) fully adsorbed (*adsorption full*). These chemicals could also be parameterized with an adsorption isotherm if relevant.

Adsorption reaction

The defined adsorption reaction calculates the equilibrium between the dissolved chemical (*classic-eq*) and the sorbed chemical (*classic-neq*) following the defined Freundlich isotherm and an adsorption rate. The equilibrium isotherm and adsorption rate are user defined and can be changed. Additionally, the adsorption reaction can be parameterized with a desorption rate if the transformation rate from sorbed to solute differs from the transformation rate from solute to sorbed (*adsorption_rate*).

The two forms of the chemical and the adsorption reaction are defined as one *chemistry* that *trace* the chemicals and simulates the sorption *reaction*.



```
(defchemical "classic-eq" classic
(adsorption none))

(defchemical "classic-neq" classic
(adsorption full))

(defreaction "classic-sorption" adsorption
    (solute "classic-eq")
    (sorbed "classic-neq")
    (equilibrium Freundlich (K_OC 20 [cm^3/g]) (m 0.9 []) (C_ref 1e-6 [g/cm^3]))
    (adsorption_rate const 1 [d^-1]))

  (defchemistry "classic-sorption" default
  (reaction "classic-sorption")
  (trace "classic-eq" "classic-neq"))
```

Figure 6.2: Example of time dependent sorption

The change in sorbed and dissolved chemical over time is calculated as a change in concentration over time. The change in chemical concentration over time (dC/dt) due to sorption is thus calculated as the difference between the equilibrium concentration (C_{eq}) and the actual concentration multiplied with a sorption (or desorption) rate.

$$\frac{\mathrm{d}C}{\mathrm{d}t} = k(C_{eq} - C) \tag{6.19}$$

where:

k = the adsorption (and/or desorption rate) [d^{-1}] (OBS: there is no default unit of k thus be aware that you specify it with a per time unit (second, hour, day etc.)

C_{eq} = the equilibrium concentration [g cm^{-3}] given by M-A_{eq}, where M is the total mass [g] and A_{eq} is given by one of the equilibrium isotherms (or equal to M if equilibrium is full).

The mass adsorbed via the time dependent adsorption reaction, Γ_{At}^{S}, can be calculated as:

$$\Gamma_{At}^{S} = \theta_t \left(\frac{\partial C_t}{\partial t}\right) \tag{6.20}$$

where:

θ_t = the volumetric water content [cm3 cm-3] at time, t, and

C_t = the concentration [g cm^{-3}] at time, t.

The Γ_{At}^{S} term is used as a sink-source term when calculating solute transport, meaning sorption and transport are decoupled.

6.7.4 Multiple levels of sorption

A chemical can be defined with multiple sorption types at the same time, both instantaneous and time-dependent, for example, weak/fast and strong/slow. All sorption rates are calculated relative to the solute concentration in the water.

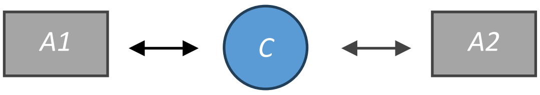


Figure 6.3: Multiple sorption types for the same chemical

In Figure 6.3 the two sorbed forms *A1* and *A2* never interact directly, only indirectly through the solute concentration, *C*. The equilibrium between *A1* and *C* and *A2* and *C* is calculated in parallel for the same *C*. There is no limit to the number of sorbed forms.

6.8 Transport in primary and secondary domain

To account for the different transport velocities, and the following different solute concentrations, in different parts of the pore system, the soil can be divided in different domains. In Daisy the soil can be divided into three domains: the primary domain, a secondary domain, and a tertiary domain. The tertiary domain accounts for water flow and solute transport in biopores and are described in section 6.9. The secondary domain accounts for solute transport in the "fast" part of the soil matrix-pore-system, for example in cracks and fissures.

Two models can be used to describe the secondary domain: The *pressure* model and the *crack* model (the *alpha model* also listed in the reference manual is just a shared base model that cannot be applied directly). Additionally, a *none* model is defined as default. Thus, there is always, per default, full equilibrium between solute in different size matrix pores. In the *pressure* model it is assumed that the hydraulic conductivity in the secondary domain can be described as part of the hydraulic conductivity curve for the entire soil matrix. Contrary, in the *crack* model it is assumed that the hydraulic conductivity of the crack models is defined by the density of the cracks.

Assumption behind the pressure model

It is assumed that the water flow and solute transport in the secondary domain can be described as part of the water flow and solute transport in the soil matrix. Thus, Richards' equation and the advection-dispersion equation can be used to described water flow and solute transport in the secondary domain. To meet this assumption, it is a prerequisite that (Søren Hansen et al., 2012):

- The applied retention curve describes the water retention in both the primary domain (matrix) and secondary domain (cracks and fissures).
- The applied hydraulic conductivity curve represents the hydraulic conductivity in both the primary domain (matrix) and secondary domain (cracks and fissures).

The pressure model

The pressure model is defined with an exchange rate between the primary (slow flowing) and secondary (fast flowing) water (*alpha* [h^{-1}]) and a pressure defining when the secondary domains are activated (*h_lim* [cm]). Additionally, a water conductivity for when the secondary domain is active, K_{lim}, can be specified (*K* [cm h^{-1}]). See more on how the parameters are used to calculate water flow and solute transport in the two domains below.

Assumption behind the crack model

Alternatively, it may be assumed that the secondary domain can be described as a simplified system of cracks with a single characteristic aperture, corresponding to one single characteristic pressure potential, h_a [cm]. Further, the crack density needs to be sufficient, so the hydraulic conductivity curve for both the primary and secondary domain can be represented by a bimodal function with a substantial increase in conductivity at h_a. If the cracks- have a dominant flow direction, this can be described with an anisotropy factor (see Chapter 4).

The cracks model

In the cracks model *h_lim* and K are calculated based on the density of cracks and the average distance between walls in the cracks. Thus, *h_lim* [m] is calculated as:

$$h_{lim} = 2 \cdot \gamma_w \frac{\cos(\alpha)}{\rho_w \cdot g \cdot a} \tag{6.21}$$

where:

γ_w = the surface tension of water, default 7.28 ·10^{-2} [N m^{-1}].

ρ_w = the density of water, default 998 [kg $m^{-3]}$].

g = gravitational acceleration, default 9.82 [m s^{-2}].

a = average distance between walls in the cracks, defined by the parameter *aperture* [m].

K [m s^{-1}], is calculated based on Poiseuille's law as:

$$K = \frac{\rho_w \cdot g \cdot a^3}{12 \cdot \eta_w \cdot \frac{1}{d_c}} \tag{6.22}$$

where:

η_w = the viscosity of water at 20 °C, default 1.00·10^{-3} [N s m^{-2}], and

d_c = the density of cracks defined by the parameter *density* [m^{-1}].

In addition, both models have the parameter *use_secondary* which as default is TRUE. Thus, when the models are specified for the secondary domains the soil matrix transport is divided into two domains for solute transport (see below). If set to FALSE the cracks will only affect the conductivity curve.

6.8.1 Water content and flow in two domains

Water content in primary and secondary domain

Based on the above assumptions and parameterization the solute transport can be described by dividing the soil matrix water, as solved by Richard's equation (see Chapter 4), in a primary and a secondary domain (Mollerup, 2010):

$$\theta = \theta_1 + \theta_2 \tag{6.23}$$

where θ_1 and θ_2 [cm^3 cm^{-3}] are the water content in the primary and secondary domain, respectively. The primary domain, representing the water content and flow in the smallest pores, is always filled first. When the matrix water content, θ, exceed a certain limit, θ_{lim}, the secondary domain begins to be filled, i.e. θ_{lim} is the maximum value of θ_1. Contrary, as the secondary domain represents the water content in the largest pores (cracks and fissures) it is emptied first. Thus, the primary domain and the secondary domain can be expressed as:

$$\begin{aligned} \theta_1 &= min(\theta, \theta_{lim}) \\ \theta_2 &= max(0, \theta - \theta_{lim}) \end{aligned} \tag{6.24}$$

Water fluxes in primary and secondary domain

The water fluxes, q [cm h^{-1}], calculated by Darcys equation (see Chapter 4) are also divided in two parts: a part representing the water fluxes in the primary domain, q_1, and a part representing the water fluxes in the secondary domain, q_2:

$$q = q_1 + q_2 \tag{6.25}$$

Similarly, is the hydraulic conductivity of the matrix soil divided in a primary and a secondary domain:

$$K = K_1 + K_2 \tag{6.26}$$

where K [cm h^{-1}] as default in the *pressure* model is the hydraulic conductivity function used for water movement calculations and defined under horizon (see Chapter 4). K_1 [cm h^{-1}] is calculated using the hydraulic conductivity function used for the water movement calculations, but with θ_1 instead of θ. Thus:

$$K_1 = K(\theta_1) \tag{6.27}$$

and K_2 [cm h^{-1}] is calculated as:

$$K_2 = \begin{cases} 0 & for\ \theta_2 = 0 \\ K(\theta) - K(\theta_{lim}) & for\ \theta_2 > 0 \end{cases} \tag{6.28}$$

Alternatively, in the cracks model and if K is specified in the *pressure* model:

$$K_2 = \begin{cases} 0 & for\ \theta_2 = 0 \\ K_{cracks} & for\ \theta_2 > 0 \end{cases} \tag{6.29}$$

The water fluxes in q_1 and q_2 can then be calculated as:

$$q_1 = \min\left(0, \frac{K_1(\theta_{lim})}{K(\theta)}\right) \cdot q \tag{6.30}$$

$$q_2 = q - q_1 = \min\left(0, \frac{K_2(\theta_{lim})}{K(\theta)}\right) \cdot q \tag{6.31}$$

The mean velocity in the pore system of the primary and secondary domain is then given by:

$$v_1 = \frac{q_1}{\theta_1} \quad \text{and} \quad v_2 = \frac{q_2}{\theta_2} \tag{6.32}$$

It should be noted that when there is water flow in the secondary domain, the velocity, v_2 [cm h^{-1}] is often considerably larger than v_1 [cm h^{-1}].

6.8.2 Solute concentration and flow in two domains

Solute concentrations in and exchanges between primary and secondary domain

As for water content, the solute concentration, C [g cm^{-3}], is divided in the secondary and primary domain:

$$C = C_1 + C_2 \tag{6.33}$$

The exchange between the two domains is driven by the concentration difference and described as a simple linear process:

$$\Gamma_{d1\to d2} = -\Gamma_{d2\to d1} = \begin{cases} \alpha_{1\to 2}(C_1 - C_2) & for\ C_1 \geq C_2 \\ \alpha_{2\to 1}(C_1 - C_2) & for\ C_1 < C_2 \end{cases} \tag{6.34}$$

where:

$\Gamma_{d1\to d2}$ and $\Gamma_{d2\to d1}$ [g cm^{-3} h^{-1}] are the transfer of solutes from the primary to the secondary domain and from the secondary to the primary domain, respectively, included as a sink-source term in the advection-dispersion equation.

$\alpha_{1\to 2}$ and $\alpha_{2\to 1}$ [h^{-1}] are exchange-coefficients for transfer from the primary to the secondary domain and from the secondary domain to the primary domain, respectively. These have the same value in Daisy and are defined by the parameter *alpha* [h^{-1}].

Solute transport in the primary domain

Solute transport in the primary domain is calculated with the advection-dispersion equation only taking into account the water content, solute concentration and velocity of the primary domain. Here shown for a non-sorbing tracer in 1-Dimension (for sorbing chemicals in primary and secondary domains see section 6.8.5):

$$\frac{\partial(\theta_1 C_1)}{\partial t} = \frac{\partial}{\partial z}\left(\theta D \frac{\partial C_1}{\partial z} - q_1 C_1\right) + \Sigma\Gamma \tag{6.35}$$

Where the sum of the sinks and source terms, $\Sigma\Gamma$, includes the exchange with the secondary domain.

Solute transport in the secondary domain

Solute transport in the secondary domain is only calculated by advection as the movement by advection is relatively large compared to the movement by molecular diffusion when $\theta_2 \neq 0$.

$$\frac{\partial(\theta_2 C_2)}{\partial t} = \frac{\partial}{\partial z}(q_2 C_2) + \Sigma\Gamma \tag{6.36}$$

6.8.3 Initial concentrations in two domains

By default, the initial concentration is zero, except for N (see chapter 7), however initial concentrations of any relevant solute in the soil matrix can be specified (see section 6.4.3). When simulating solute transport in a primary and secondary domain the initial concentration in each domain can be specified using the *initial_C_secondary* and *intital_C_primary* or the *initial_C_secondary_plf* and *initial_C_secondary_plf*. Alternatively, the concentration in the secondary domain may be given by the optional state variable *C_secondary*. For the difference between the parameters see section 6.4.3 and the parameter overview. If the initial concentration is defined for the total soil matrix (with for example *initial_C*, see section 6.4.3), both the primary and secondary domain will be initialized with the specified concentration.

Initial mass

As an alternative to solute concentration in the soil water, the initial amount of a chemical can be specified as the total mass per volume water, soil, and air defining the primary and secondary domain, respectively, by the parameters *initial_M_secondary* and *initial_M_primary* or *initial_M_secondary_plf* and *initial_M_primary_plf*. For the difference between these parameters see section 6.4.3 and the parameter overview. If the initial mass is defined for the total soil matrix (with for example *initial_M*, see section 6.4.3), both the primary and secondary domain will be initialized with the specified concentration.

6.8.4 The sink terms in two domains

Transport to drains

By default, solute transport to drains, when simulating with a primary and secondary domain, is calculated as the average concentration of the primary and secondary domain:

$$\Gamma^{s}_{drain} = \frac{\theta_1 \cdot C_1 + \theta_2 \cdot C_2}{\theta_1 + \theta_2} \tag{6.37}$$

where Γ_{DF} [cm^3] is water flow to drains (see Chapter 4). It would be more physically correct to use the concentration in the secondary domain, as this characterizes the highly mobile water and solute, which would flow to the drains. This can be obtained by setting the parameter *drain_secondary* to TRUE (under *chemical*), however it may give unstable results.

Groundwater leaching

Solute transport out of the lower boundary of the soil column (e.g. estimate of groundwater leaching) is simply calculated as the sum of solute transport over the lower boundary calculated for the primary and secondary domain, respectively.

Root uptake

The root uptake of chemicals when simulating two domains is calculated like root uptake from one domain (Eq. 15), but with preferential uptake from the secondary domain. Thus, if the water content in the secondary domain is sufficient to supply the root with water, C in eq. 6.15 will be equal to C_2. Alternatively, C will be calculated as a weighted average between C_1 and C_2 as done for transport to drains (Eq. 6.37) but including all available water in θ_2.

6.8.5 Sorption in two domains

Adsorption and desorption are calculated for the chemical content in the two domains separately. Thus, adsorption in the primary domain is a function of sorption sites and solute concentration in the primary domain and similarly, adsorption in the secondary domain is a function of sorption sites and solute concentration in the secondary domain. The solute concentration is given by the advection-dispersion equation describing solutes transport in the two domains respectively (see section 6.8.2).

The primary domain accounts for water flow and solute transport in the smallest soil pores. Thus, the amount of available sorption sites in the primary domain is higher pr. volume of water compared to the secondary domain. The sorption equilibrium for the two domains is calculated as:

$$\begin{aligned} S_{eq1} &= sf_1 * S_{eq} \\ S_{eq2} &= (1 - sf_1) * S_{eq} \end{aligned} \qquad (6.38)$$

where:

S_{eq} = the equilibrium sorbed chemical [g g^{-1}] calculated by one of the equilibrium isotherms (or fully sorbed) (see section 6.7) and

sf_1 = a primary sorption fraction [-]

The primary sorption fraction is calculated as the fraction of equivalent pore size space in the primary domain over the total equivalent pore size space in the soil. These are calculated by integrating the pressure potential, h over the volumetric soil water content, θ, in the area between θ_{min} and θ_{lim} and between θ_{min} and θ_{sat}, respectively:

$$\begin{aligned} ss_1 &= \int_{\theta_{min}}^{\theta_{lim}} h(\theta) d\theta \\ ss &= \int_{\theta_{min}}^{\theta_{sat}} h(\theta) d\theta \end{aligned} \qquad (6.39)$$

where:

θ_{min} = the water content in the smallest pores in the soil defined by the *r_pore_min* parameter given under horizon (default 0.1 µm, which corresponds to h = -15000 cm).

6.9 Transport in biopores

Solute transport in the tertiary domain

Daisy allows for an advanced and a simple description of biopores, accounting for solute transport in the tertiary domain. The geometry and water flow in both the advanced and simple model are described in Chapter 4. In both models we assume that advection dominates the solute transport as vertical transport in biopores is considered much faster than the lateral transport into the soil matrix. Thus, the tertiary component (under movement) can be defined by either *biopores* (the advanced biopore model), *old* (the simple default biopore model), or *none* (no biopores).

In the simple model, water and solute are transferred directly to the matrix soil where the biopores terminate. Contrary, for the advanced model, where the biopores have a water holding capacity, solute can be contained in the biopore. Complete mixing of the solute in the biopore is assumed. Despite the potential residence time in the biopores sorption to the biopore wall, dispersion over the biopore-matrix interface and degradation of the solute in the biopores is neglected.

The simple biopore model

In the simple biopore model solute transport is solely calculated by advection. Thus, the sink-source term, both for the soil supplying the biopore with water and solute (topsoil) and for the soil receiving water and solute through the biopore (subsoil), is a result of the amount of incoming water flow and the concentration in the water, calculated as:

$$\Gamma_{Bf}^{s} = \Gamma_{Bf} C \tag{6.40}$$

where:

Γ_{Bf}^{s} = the sink-source term for solute transport through biopores [cm^3 cm^{-3} h^{-1}].

Γ_{Bf} = water flow through the biopores [cm^3 cm^{-3} h^{-1}].

C = solute concentration in the matrix water where the water flow through the biopores originates from [g m^{-3}].

The simple biopore model is by default enabled if the combined amount of humus and clay in the top horizon is above 5 %.

The simple biopore model is termed *old* and takes two parameters *macro* and *mactrans* which by default are specified with the *macro* and *mactrans* models that are used to calculate preferential water flow and preferential solute transport, respectively, as described above.

The advanced biopore model

In the advanced biopore model water and solute can build up in the biopore. For solute in the biopore we assume complete mixing, thus the sink-source term for solute transport through the biopore is calculated as:

$$\Gamma_{Bf}^{s} = \Gamma_{Bf}^{in} C + \Gamma_{Bf}^{out} C_b \tag{6.41}$$

where:

Γ_{Bf}^{in} and Γ_{Bf}^{out} are the positive (water going from the matrix to the biopore) and negative (water going from the biopore to the matrix) water sink terms, respectively [m^3 m^{-3} h^{-1}] and

C_b is the concentration of solute in the biopore [g cm^{-3}]. C_b is given by:

$$C_b = \frac{M_b}{s} \tag{6.42}$$

where:

s = the water storage in each biopore (see Chapter 4) [cm] and

M_b = the mass of solute in the biopore per unit area [g cm^{-2}].

The change in M_b over time is calculated as:

$$\frac{dM_b}{dt} = q_{Pond,inf}^{b} \cdot C_{pond,inf} + \int_{z_b^{bot}}^{z_b^{top}} \Gamma_{Bf}^{s} dz \tag{6.43}$$

where:

$q_{Pond,inf}^{b}$ = ponding water infiltrating the biopores [cm h^{-1}] (see Chapter 4) and

$C_{pond,inf}$ = the concentration of the solute in the ponding water [g cm^{-3}]

6.10 Colloids

Colloid-facilitated transport of chemicals happens when a sorbing chemical adsorbs to a soil-particle (a colloid) that is then detached from the soil matrix and transported with the waterflow through the soil. Colloid facilitated transport is controlled by the following sub-processes (Hansen et al., 2012):

1. Generation of colloids
2. Adsorption to the colloid
3. Transport of mobilised colloids
4. Filtration of the mobilised colloids.

6.10.1 Generation of colloids

Colloid generation on the surface is a function of rainfall erosivity (related to kinetic energy or momentum of drops), rainfall erodibility, and protecting elements such as a plant cover, litter/mulch, or standing water.

Two different models are implemented to estimate kinetic energy, three different models are available to describe the reducing effect of a water layer (e.g., ponding) on splash detachment, and three different models for the generation of colloids at the surface are available:

- A method based on kinetic energy, but modified according to the MACRO model (Jarvis et al., 1999), (Jarvis and Larsson, 1998).
- A method based on the momentum of the drops (Styczen and Høgh-Schmidt, 1988), and
- A method based on the kinetic energy of rainfall (Morgan et al., 1998),

All three models consider that litter and mulch reduce the splash detachment, but they describe the effect of a canopy quite differently.

The two implemented models to estimate kinetic energy are based on the Universal Soil Loss Equation and the EUROSEM erosion model, respectively.

Kinetic Energy, Brown and Foster, 1987

Energy of rainfall can be calculated according to the revised Universal Soil Loss Equation (Brown and Foster, 1987):

$$E = 29 \cdot (1 - 0.72 \exp(-0.05 \cdot P_r)) \tag{6.44}$$

where:

P_r = rainfall rate [mm h^{-1}] and

E = kinetic energy [J m^{-2} mm^{-1}]

Total kinetic energy is then $KE_{DT} = E \cdot direct\ rain$ [J m^{-2} h^{-1}], where *direct rain* includes rain hitting ponded water or litter, but excludes rain hitting canopy or snow, as well as snow and all forms of irrigation. This method is by default used with the splash detachment model by Jarvis and Larsson (1998), implemented in the submodel “colgen_Jarvis99” and similar to the implementation in the MACRO-model. Thus, in this implementation, an area covered by canopy has no kinetic energy reaching the surface.

Kinetic Energy, Brandt (1989)

Kinetic energy can also be calculated as it is done in the EUROSEM erosion model. The total kinetic energy from the rainfall stems from rainfall hitting the soil directly and energy from rainfall dripping off leaves:

$$KE_{Tot} = KE_{DT} + KE_{LD} \tag{6.45}$$

where:

KE_{DT} = Kinetic energy of drops falling directly on the ground [J cm^{-2} h^{-1}]

KE_{LD} = Kinetic energy from leaf drip [J cm^{-2} h^{-1}].

Energy of direct rainfall is estimated as a function of rainfall intensity developed by Brandt (1989):

$$KE_{DT} = (8.95 + 8.44 \cdot \log(i)) \cdot J_{w,d} \tag{6.46}$$

where:

I = rainfall intensity [mm hr^{-1}]

$J_{w,d}$ = direct rainfall = throughfall, eq. 3.17 in Chapter 3.

The energy of leaf drip is estimated by Brandt (1989) to be:

$$KE_{LD} = \left(15.8 \cdot H_{plant}^{0.5} - 5.87\right) \cdot J_{w,C} \tag{6.47}$$

where:

H_{plant} = effective plant height [m]

$J_{w,C}$ = drip-off from the plant canopy, [mm hr^{-1}], eq.3.22 in Chapter 3)

The kinetic energy is set to 0, when the vegetation is lower than 14 cm to avoid negative values of eq. (6.47). However, in this model, drip from the canopy contributes to splash detachment on the surface.

Water depth functions

The depth factor, K_h expresses the reduction in splash erosion with increasing water depth, h [m], of ponding on the surface due to absorption of energy by the water. Several exponential functions have been suggested to describe this reduction. In Daisy it is possible to choose between equations suggested by Park et al. (1982), Hairsine and Rose (1991), Morgan et al. (1998) and “none” ($K_h = 1$), see Figure 1.

The function by Morgan et al. (1998) (also used in the EUROSEM model) describes K_h as:

$$K_h = e^{(-b \cdot h)} \tag{6.48}$$

where:

b = experimentally determined coefficient [mm^{-1}]. The range of “*b*” is from 0.9 to 3.1, and a default value of 2 is adopted.

The model by Park relates the depth factor to the median drop size:

$$K_h = 2.7183 \cdot e^{\left(\frac{-h}{dds}\right)} \tag{6.49}$$

where:

dds = median drop diameter [m], calculated from the empirical relationship $dds = 1.238 \cdot I^{0.182}$.

The water depth factor by Harisine and Rose resembles the expression by Park, but applies a power function:

$$K_h = \left(\frac{-h}{dds}\right)^{-0.8} \tag{6.50}$$

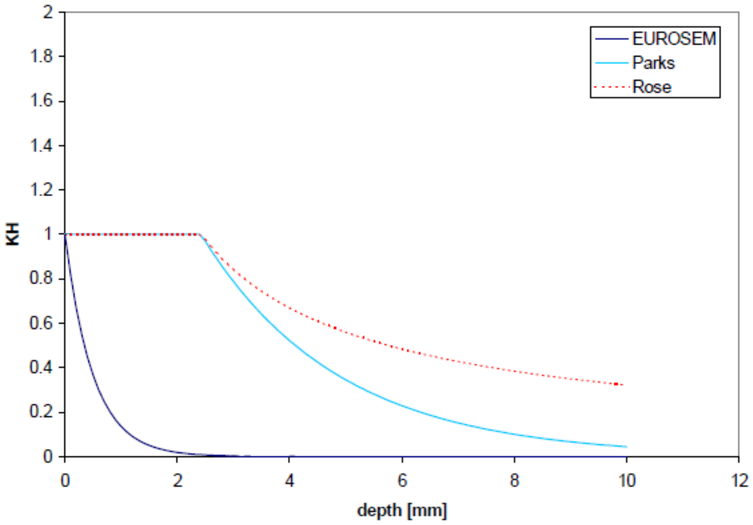


Figure 6.4: The water depth functions calculated for a rainfall intensity of 40 mm hr⁻¹.

Detachment, MACRO-model

Splash detachment of particles in the colgen_Jarvis99-submodel (similarly to splash detachment in the MACRO-model) is described as

$$D_s = K_h \cdot k_d \cdot KE_{Tot} \cdot (1 - A_M) \cdot M_s \tag{6.51}$$

where:

D_s = Splash detachment [g cm^{-2} hr^{-1}]

K_h = a water depth function (default = 1 []),

k_d = an erodibility index, [g J^{-1}]

KE_{Tot} = total kinetic energy, by default calculated by the Brown and Foster-method, eq. (1).

A_M = the litter/mulch cover fraction protecting the soil from drops [].

M_s = mass fraction of dispersible/movable particles [g (g soil)$^{-1}$]

The amount of particles (colloid size) that can be detached from M_s is time dependent, described as:

$$\rho_s \cdot z_i \frac{\partial M_s}{\partial t} = -\alpha \cdot D_s + R_{rep} \tag{6.52}$$

where:

ρ_s = bulk density of the soil [g m^{-3}],

z_i = depth of topsoil influenced by detachment and dispersion [m],

α = the faction of particles which are transported away from the surface ($0 \leq \alpha < 1$) [],

R_{rep} = the rate at which the amount of particles which can be detached are reproduced [g m^{-2} s^{-1}].

The process of reproduction of particles for detachment is described as:

$$R_{rep} = k_r \cdot \left(1 - \frac{M_s}{M_{max}}\right) \tag{6.53}$$

where:

k_r = rate of particle reproduction [g m^{-2} s^{-1}],

M_{max}= the maximum amount of colloids available [g g soil^{-1}].

M_{max} is, by default calculated as M_{max} = 0.362 * clay [%] – 0.00518 (Brubaker et al., 1992). Experience has shown that M_{max} may vary over time. It is possible to modify the value as a function of days after tillage. The fraction to multiply with may be specified as a piecewise linear function (plf).

Detachment, Morgan (1998)

The detachment of soil particles by splash erosion is described as in the EUROSEM model (Morgan et al., 1998):

$$D_s = K_h \cdot k_d \cdot KE_{Tot} \cdot (1 - A_M) \tag{6.54}$$

where:

KE_{Tot} = Total kinetic energy of the rainfall [J cm^{-2} hr^{-1}] by (Brandt, 1989), eq. 2-4.

K_d = A soil erodibility index [g J^{-1}]

K_h = a water depth factor. The default choice is eq. (5).

Effects of plant canopy are considered in the calculation of kinetic energy.

Detachment, Rainfall Momentum Model

Styczen and Høegh-Schmidt (1988) describe splash erosion as a function of the squared momentum of drops hitting directly and drops generated on leaves.

$$D_s = A(e) \cdot (1 - A_M) \cdot K_h \cdot C_M \cdot M_R \tag{6.55}$$

where:

$A(e)$ a soil resistance factor [hr^2 g^{-1} cm^{-2}] originally [s^2 kg^{-1} m^{-2}], describing the average amount of energy required to detach a particle as well as the probability that it will be moved into the water layer.

K_h one of the equations 5-7, above, chosen by the user.

C_M the total squared momentum of drops hitting the soils divided by the squared momentum of drops falling directly [],

M_R The squared momentum of drops falling directly on bare soil [g^2 hr^{-3}]. The original unit was (N s)^2 m^{-2} s^{-1}.

The squared momentum depends on the drop size distribution of the rainfall. For rainfall following the Marshall Palmer distribution, it is approximately proportional to rainfall intensity to a power [(N s)^2 m^{-2} s^{-1}]:

$$M_R = \begin{cases} (2.04 \cdot 10^{-8}) \cdot I^{1.63} & \text{for } I \ \leq 75 \text{ mm h}^{-1} \\ 4.83 \cdot 10^{-8}) \cdot I^{1.43} & \text{for } I \ > 75 \text{ mm h}^{-1} \end{cases} \tag{6.56}$$

The Canopy cover-momentum factor, CM, describes the relative effect of vegetation on soil detachment and can be described as the actual squared momentum of drops hitting the soil directly and from canopy drip-off, divided by the squared momentum calculated without vegetation cover:

$$C_M = \frac{(1 - A_C) \cdot M_R + J_{w,C} \cdot M_C}{M_R} \tag{6.57}$$

where:

A_C = fraction covered by a canopy

M_C = squared momentum of drops from the vegetation [kg^2 m^{-1} s^{-2}]

$J_{w,C}$ = Canopy drip-off. [m s^{-1} = m^3 m^{-2} s^{-1}]

MC depends on drop velocity which again depends on drop size and fall height. Drop velocities were measured by Epema and Riezebos (1983) for different combinations of drop sizes and fall height. The relationship is described as:

$$M_C = a + bH_C + cH_C^2 + dH_C^3 \tag{6.58}$$

Where the values of the constants are given in Table 1 below and H_C is the height of the canopy from which the drops fall.

Table 1. Constants required to calculate the relationship between drop size, fall height, and squared momentum of drops.

Plant		Drop sizes, mm			
height		4.5	5.0	5.5	6.0
<2 m	a	-			
	b	0.7954	1.1058	1.4916	1.9601
	c	-	-	-	-
	d	-	-	-	-
2-13 m	a	-0.5	-0.5	-0.5	-0.5
	b	1.2031	1.5930	2.0692	2.5496
	c	-0.12416	-0.15954	-0.20184	-0.23976
	d	4.33E-3	5.44E-3	6.70E-3	7.68E-3
>13 m	a	3.8647	5.4080	7.2934	9.5310
	b	-	-	-	-
	c	-	-	-	-
	d	-	-	-	-

6.10.2 Adsorption to colloids

The detached colloids are then mixed into the ponding water and water in the mixing layer, just as a solute. However, colloids can also, if so parameterized, sorb solutes to the surface, and thus create an additional transport pathway for (strongly) sorbing solutes. Sorption to the colloids is calculated as time-dependent sorption following a linear sorption isotherm (Eq. 6.19 and 6.21). The sorption of the chemical will be distributed between colloids and the soil in the surface mixing layer according to the weight fraction the colloids constitute of the mixing layer. Colloids may have a larger K_{ld}-value for sorption than the original soil, as colloids typically are made of finer material with a larger surface area. A *soil enrichment factor* [] can be set, specifying what the K_{ld} of the soil should be multiplied with, to represent K_{ld} of the colloids.

When chemicals are adsorbed to colloids, they will move through the soil profile together with the colloids, when these are transported with the water (section 6.10.3). In the soil, colloids may be filtered from the water phase (section 6.10.4).

6.10.3 Transport of colloids

Transport of colloids is calculated as for other chemicals with the advection-dispersion Equation (Eq. 6.1) (see section6.3). The upper boundary is defined as a flux boundary, where the flux is given by: $\alpha \cdot D_s$ [g cm^{-2} h^{-1}], where α is the fraction of particles that are transported away from the surface ($0<\alpha<1$) [] and D_s is the splash detachment [g cm^{-2} h^{-1}] (see appendix 3.4).

If the water flux and solute transport are simulated in multiple domains (secondary and/or tertiary), equal amounts of colloids are assumed to enter each domain.

6.10.4 Filtration of colloids

The mobilised colloids are captured in the soil profile by filtration. The filtration rate, Γ_{flt} [g cm^{-3} h^{-1}], is calculated as (Hansen et al., 2012):

$$\Gamma_{flt} = f_c \cdot C_k \cdot q \tag{6.59}$$

where:

f_c = filtration coefficient [cm^{-1}],

q = water flow [cm h^{-1}] (see Chapter 4), and

C_k = colloid concentration [g cm^{-3}]

Γ_{flt} is implemented as a sink-source term in the advection-dispersion equation used to calculate colloid transport. If water flow and solute transport are calculated for both the primary and secondary domain, Eq. 6.59 is calculated separately for the two domains based on the water flow and colloid concentration in the respective domain. In this case, f_c can be specified for each domain.

Filtration of colloids is not calculated in the tertiary domain (biopores).

When colloids are filtrated, the chemicals adsorbed to the colloids will be adsorbed to the soil matrix. Contrary, when colloids are generated, chemicals are moved from the soil matrix to colloids. The amount of replaced chemicals follows the amount of generated and filtrated colloids. Thus, if 10 % of the colloids are filtrated, 10 % of the relevant chemical adsorbed to the colloids will be adsorbed to the soil matrix, and if 1 % of the soil matrix is detached as colloids, 1 % of the relevant chemical will change from matrix bound to colloid bound.

By default colloids (and adsorbed chemical) will not be traced as a colloid after filtration, however this can be change by specifying the *immobile* parameter of *filter_velocity* model.

6.11 Numerical solutions

1D numerical solution

The numeric solution to the convection-dispersion equation in 1D (the *Hansen model*) is based on Wagenet and Hutson (1989).

2D numerical solution

The numeric solution to the convection-dispersion equation in 2D (the *Mollerup model* (Mollerup, 2010)) is found with the *exparse solver*, which uses the CXSparse library (Davis, 2006). Alternatively, the *ublas solver,* using the UBLAS lu functions provided by Gunter Winkler (guwi17@gmx.de), can be applied.

As default at full upstream formulation is applied (*upstream_weight* 1), thus the concentration in the water flux over cell boundaries is equal to the concentration in the cell the water is coming from. Additionally, diffusion over boundaries is, by default enabled (*enable_boundary_diffusion* TRUE).

Two stabilizing methods can be applied: *reduce_timestep* or *streamline_diffusion* (increases diffusion). As default *reduce_timestep* is applied. If it is wished not to apply a stabilizing method, *stabilizing_method* may be parameterized with *none*.

6.12 Parameter overview

Table 6.2. Related Parameter names in Daisy.

Name and explanation		Model (in Daisy)	Parameter name (Daisy reference manual)	Default	Default unit
Trace	List of chemicals that should be traced.	default (chemistry)	*trace*	User specified	[sequence]
		DOM (chemistry)	*trace*	DOC, DON	[sequence]
		N (chemistry)	*trace*	N	[sequence]
Reaction	List of chemical reaction that should be calculated.	default (chemistry)	*reaction*	User specified	[sequence]
		DOM (chemistry)	*reaction*	DOM_turnover	[sequence]
		N (chemistry)	*reaction*	nitrification, denitrification	[sequence]
Combine	List of chemistry parameterizations that should be combined.	multi (chemistry)	*combine*	User specified	[sequence]
		nutrient (chemistry)	*combine*	N and user specified	[sequence]
Ignore	List of chemicals included in combine that are not traced.	multi (chemistry)	*ignore*	User specified	[sequence]
Max_sink_total	Maximum allowed sink term as a fraction of total content.	multi (chemistry)	*max_sink_total*	Default 0.5	[]
Max_sink_solute	Maximum allowed sink term as a fraction of solute content.	multi (chemistry)	*max_sink_solute*	Default 0.9	[]

Max_sink_secondary	Maximum allowed sink term as a fraction of secondary domain content. This should usually be above 1 to allow for the case where the secondary domain is emptied within a timestep.	multi (chemistry)	*max_sink_secondary*	Default 1.5	[]
Min_sink_total	Always allow this fraction of total content to be removed by sink term. This overwrites all the 'max_sink' parameters.	multi (chemistry)	*Min_sink_total*	Default 0.01	[]
Decomposition rate	How fast the chemical is being decomposed in the soil. Must be specified with either decompose rate or decompose halftime. decompose_rate = ln(2)/decompose_halftime	base (chemical)	*decompose_rate*	User specified	[h^{-1}]
Decompose halftime	How fast the chemical is being decomposed in the soil. Must be specified with either	base (chemical)	*decompose_halftime*	User specified	[h]

	decompose rate or decompose halftime.				
C	Solute concentration in the water.	base (chemical)	*C*	Optional state variable	[g cm^{-3}]
D_0	The diffusion coefficient	base (chemical)	*Diffusion_coefficient*	User specified	[cm^{2} s^{-1}]
τ	The tortuosity calculated by either a linear model (default) or using the Millington-Quirk approach (M_Q).	Horizon	*tortuosity*	Default linear.	[-]
a	Theta offset.	linear (tortuosity)	*a*	Default = wilting point	[cm^{3} cm^{-3}]
b	Theta factor	linear (tortuosity)	*b*	Default 2	[-]
λ	Dispersion length	Soil (fixed components)	*dispersivity*	Default 5	[cm]
	Transversal dispersion length		*Dispersivity_transversal*	Default 0.1 times dispersivity	[cm]
C_below	Concentration below the simulated soil column. Default is the zero_gradient component that assumes a concentration equal to the concentration in the lowest cell.	base (chemical)	*C_below*	Default zero_gradient Alternatively, a positive concentration or, a negative number which indicate same concentration as in lowest cell (equal to zero gradient)	[const model] [User defined]
Initial	The initial chemical content. Default zero.	base (chemical)	*initial*	Default 'initial_zero'	[g cm^{-3}]

Initial_C	The initial solute concentration in the water. Defined from top going down.	base (chemical)	*initial_C*	User specified	plf pairs [END [cm] VALUE [g cm^{-3}]
Initial_C_plf	The initial solute concentration in the water. Defined from bottom going up.	base (chemical)	*initial_C_plf*	User specified	plf pairs [DEPTH [cm] VALUE [g cm^{-3}]
	Total mass per volume water, soil and air.	base (chemical)	M	Optional state variable	[g cm^{-3}]
Initial_M	The initial mass per volume water, soil and air. Defined from top going down.	base (chemical)	*initial_M*	User specified	plf pairs [END [cm] VALUE [g cm^{-3}]
Initial_M_plf	The initial mass per volume water, soil and air. Defined from bottom going up.	base (chemical)	*initial_M_plf*	User specified	plf pairs [DEPTH [cm] VALUE [g cm^{-3}]
chemical	The name of the chemical to be removed from the field.	remove_solute (action)	*chemical*	User specified	[-]
solute	Solute in irrigation water specified with name of chemical and concentration (value) of chemical.	irrigate_base (action)	*name* *value*	User specified User specified	[] [ppm]
S_permanent	Permanent external source	base (chemical)	*S_permanent*	User specified	[g cm^{-3} h^{-1}]

rc	How much of the chemical is reflected at crop uptake. 1 = full uptake, 0 = no uptake.	base (chemical)	*crop_uptake_reflection_factor*	Default 1	[fraction]
ρ	Dry bulk density	Horizon	*Dry bulk density*	User specified	[g cm^{-3}]
K_{fclay}	Clay dependent distribution parameter.	Freundlich (adsorption)	*K_clay*	User specified	[cm^{3} g^{-1}]
K_{fOC}	Organic carbon dependent distribution parameter.	Freundlich (adsorption)	*K_OC*	User specified	[cm^{3} g^{-1}]
m	Freundlich exponent.	Freundlich (adsorption)	*m*	User specified	[-]
C_{ref}	Reference concentration. Recommended value 1e^{-6}.	Freundlich (adsorption)	*C_ref*	User specified	[g cm^{-3}]
K_{l}	The Langmuir half saturation constant.	Langmuir (adsorption)	*K*	User specified	[g cm^{-3}]
a_{max_clay}	The max adsorption capacity for clay.	Langmuir (adsorption)	*my_max_clay*	User specified	[g cm^{-3}]
a_{max_OC}	The max adsorption capacity for organic carbon.	Langmuir (adsorption)	*my_max_OC*	User specified	[g cm^{-3}]
K_{ld}	The soil dependent distribution parameter. If not specified it will be calculated from K_{clay_l} and K_{OC_l}.	Linear (adsorption)	*K_d*	User specified	[cm^{-3} g]

$\boldsymbol{K_{lclay}}$	The clay dependent distribution parameter.	Linear (adsorption)	*K_clay*	User specified	[cm^{-3} g]
$\boldsymbol{K_{lOC}}$	The organic carbon dependent distribution parameter.	Linear (adsorption)	*K_OC*	User specified	[cm^{-3} g]
	Name of the solute form of the chemical.	adsorption (reaction)	*solute*	User specified	[]
	Name of the sorbed form of the chemical	adsorption (reaction)	*sorbed*	User specified	[]
	Function for calculating equilibrium between solute and sorbed form.	Adsorption (reaction)	*Equilibrium*	User specified	[]
$\boldsymbol{k}$	Transformation rate from solute to sorbed form.	adsorption (reaction)	*adsorption_rate*	User specified	[]
	Transformation rate from sorbed to solute. By default, equal to adsorption rate.	adsorption (reaction)	*desorption_rate*	User specified	[]
$\boldsymbol{\alpha_{1\rightarrow2}}$ **and** $\boldsymbol{\alpha_{2\rightarrow1}}$	Exchange coefficient for transfer between the primary and secondary domain.	alpha (secondary)	*alpha*	User specified	[h^{-1}]
$\boldsymbol{K_{crack}}$	Water conductivity when the secondary domain is active. When 0 K is calculated from	pressure (secondary)	*K*	Default 0	[cm h^{-1}]

	the hydraulic conductivity curve.				
h_lim	Pressure for activation of the secondary domain	pressure (secondary)	*h_lim*	User specified	[cm]
d_c	Density of cracks	cracks (secondary)	*Density*	User specified	[m^{-1}]
a	Average distance between walls in cracks	cracks (secondary)	*Aperture*	User specified	[m]
Use secondary	Use two domains to calculate solute transport in matrix. If FALSE cracks will only affect the conductivity curve.	cracks (secondary) and pressure (secondary)	*use_secondary*	Default TRUE	[]
C_secondary	The solute concentration in the secondary domain	base (chemical)	*C_secondary*	Optional state variable	[g cm^{-3}]
initial_C_primary	The initial solute concentration in the water of the primary domain. Defined from top going down.	base (chemical)	*initial_C_primary*	User specified	plf pairs [END [cm] VALUE [g cm^{-3}]
initial_C_primary_plf	The initial solute concentration in the water of the primary domain. Defined from bottom going up.	base (chemical)	*initial_C_primary_plf*	User specified	plf pairs [DEPTH [cm] VALUE [g cm^{-3}]

initial_C_secondary	The initial solute concentration in the water of the secondary domain. Defined from top going down.	base (chemical)	*initial_C_secondary*	User specified	plf pairs [END [cm] VALUE [g cm^{-3}]
initial_C_secondary_plf	The initial solute concentration in the water of the secondary domain. Defined from bottom going up.	base (chemical)	*initial_C_secondary_plf*	User specified	plf pairs [DEPTH [cm] VALUE [g cm^{-3}]
initial_M_primary	The initial mass per volume water, soil and air in the primary domain. Defined from top going down.	base (chemical)	*initial_M_primary*	User specified	plf pairs [END [cm] VALUE [g cm^{-3}]
initial_M_primary_plf	The initial mass per volume water, soil and air in the primary domain. Defined from bottom going up.	base (chemical)	*initial_M_primary_plf*	User specified	plf pairs [DEPTH [cm] VALUE [g cm^{-3}]
initial_M_secondary	The initial mass per volume water, soil and air in the secondary domain. Defined from top going down.	base (chemical)	*initial_M_secondary*	User specified	plf pairs [END [cm] VALUE [g cm^{-3}]
initial_M_secondary_plf	The initial mass per volume water, soil and air in the secondary	base (chemical)	*initial_M_secondary_plf*	User specified	plf pairs [DEPTH [cm] VALUE [g cm^{-3}]

	domain. Defined from bottom going up.				
drain_secondary	If true concentration in the secondary domain is used to calculate concentration in drain water. If false an average of the concentration in the primary and secondary domain is used.	base (chemical)	*drain_secondary*	Default false	[]
r_pore_min	Size of the smallest pores in the soil	Horizon	*r_pore_min*	Default 0.1	[µm]
immobile	Immobile colloids in the soil. By default, filtered colloids are not tracked.	filter_velocity (reaction)	*Immobile*	Optional parameter	string
	Mobile colloids dissolved in soil water.	filter_velocity (reaction)	*mobile*		string
f_c	Filter coefficient in the primary domain.	filter_velocity (reaction)	*fc_primary*	User defined	[cm^{-1}]
f_c	Filter coefficient in the secondary domain.	filter_velocity (reaction)	*fc_secondary*	User defined	[cm^{-1}]

$\boldsymbol{K_{ld}}$	The soil dependent distribution parameter. If not specified it will be calculated from K_{clay_l} and K_{OC_l}.	sorption (reaction)	*K_d*	User specified	[cm^{-3} g]
	The clay dependent distribution parameter.	sorption (reaction)	*K_clay*	User specified	[cm^{-3} g]
	The organic carbon dependent distribution parameter.	sorption (reaction)	*K_OC*	User specified	[cm^{-3} g]
	Name of the solute form of the chemical.	sorption (reaction)	*solute*	User specified	[]
	Name of the sorbed form of the chemical	sorption (reaction)	*sorbed*	User specified	[]
	Name of the colloids	sorption (reaction)	*colloid*	User specified	[]
	Transformation rate from solute to sorbed form.	sorption (reaction)	*k_sorption.*	User specified	[]
	Transformation rate from sorbed to solute. By default, equal to *k_desorption.*	sorption (reaction)	*k_desorption*	By default equal to *k_soprtion.*	[]
	Represent how much more chemicals sorp to colloids instead on soil matric. Multiplied with *K_d*.	soprtion (reaction)	*soil_enrichment_factor*	Default 1	[]

b	Exponent in the expression for dampening splash erosion by water on the surface, eq. (5) (Morgan et al., 1998)	EUROSEM	*b*	2	[mm^{-1}]
K_h	A water depth function	colgen_Jarvis99 colgen_Morgan98 colgen_Styczen88	*ponddamp*	1 (the option “none”) EUROSEM optional	[]
k_d	Detachment rate coefficient	colgen_Jarvis99 colgen_Morgan98	*kd*	User specified	[g J^{-1}]
M_s	Current concentration of detachable particles in the topsoil	colgen_Jarvis99	*Ms*	10 % of *Mmax* (for initialization)	
M_{max}	Maximum amount of detachable particles	colgen_Jarvis99	*Mmax*	M_{max} = 0.362 * clay [%] – 0.00518 (Brubaker et al., 1992)	[g g^{-1}]
	Sets *Ms=Mmax* after tillage	colgen_Jarvis99	*Tillage_replenish all*	false	
	Modifier to *Mmax* over time	colgen_Jarvis99	*Mmax_tillage_factor*	(Mmax_tillage_factor (0 1) (1 1))	plf [d → <none>]
k_r	Replenishment rate coefficient	colgen_Jarvis99	*kr*	User specified	[g cm^{-2} h^{-1}]
z_i	Thickness of surface soil layer	colgen_Jarvis99	*zi*	z_mixing from “Surface”	[cm]
$A(e)$	a soil resistance factor describing the average amount of energy	colgen_Styczen88	*Ae*	User specified	[hr^2 g^{-1} cm^{-2}]

	required to detach a particle as well as the probability that it will be moved into the water layer.				
A_M	Protective cover	colgen_Styczen88	*MA*	Cover predicted by the litter model	[]
dropsize	Size of droplets from vegetation	colgen_Styczen88	*Droplet_diameter*	User specified	[mm]
	Factor describing how many times higher the K_d of sorption to colloids is compared to the K_d of the soil.	sorption	*soil enrichment factor*	1	[]
	Model used for solving the matrix equation system.	Mollerup	*solver*		
debug	Number of debug messages.	Mollerup	*debug*	Default 0	[-]
Stabilizing method	Method to increase the stability of the solution to the convection-dispersion equation.	Mollerup	*stabilizing_method*	Default *timestep_reduction.* Alternatively: *streamline_diffusion* or *none.*	[-]
Enable boundary diffusion	Control if diffusion over boundaries is accounted for.	Mollerup	*enable_boundary_diffusion*	Default TRUE	[-]

Upstream weight	Upstream weighting factor.	Mollerup	*upstream_weight*	Default 1	[fraction]

Original text from	Hansen, 2002; Hansen et al., 2012; Mollerup, 2010; Mollerup et al., 2014	
Updated by	date	For Daisy version
Jakobsen, C., Holbak, M. and Styczen, M.	2025 03 13	7

6.1 References

Davis, T.A., 2006. Direct Methods for Sparse Linear Systems, SIAM Book Series on the Fundamentals of Algorithms. SIAM, Philadelphia.

FOCUS, 2012. FOCUS Surface Water Scenarios in the EU Evaluation Process under 91/414/EEC”. Report of the FOCUS Working Group on Surface Water Scenarios, EC Document Reference SANCO/4802/2001-rev.2. (No. version 1.2).

Freundlich, Heller, 1939. The adsorption of cis- and trans-azobenze. Journal of the American Chemical society 61, 2228–2230.

Hansen, S., 2002. Daisy, a flexible Soil-Plant-Atmosphere system Model.

Hansen, S., Petersen, C.T., Mollerup, M., Abrahamsen, P., Gjettermann, B., Nielsen, M.H., Styczen, M., Poulsen, R., Lørup, J.K., Yamagata, K., Butts, M., 2012. Flerdimensional modellering af vandstrømning og stoftransport i de øverste 1-2m af jorden i systemer med markdræn (No. 138), Bekæmpelsesmiddelforskning. Miljøstyrelsen.

Hansen, Søren, Petersen, C.T., Mollerup, M., Abrahamsen, P., Gjettermann, B., Nielsen, M.H., Styczen, M.E., Poulsen, R., Lørup, J.K., Yamagata, K., Butts, M.B., 2012. Flerdimensional modellering af vandstrømning og stoftransport i de øverste 1–2 m af jorden i systemer med markdræn. Miljøstyrelsen.

Millington, R.J., Quirk, J.P., 1961. Permeability of porous solids. Trans. Faraday Soc. 57, 1200–1207. https://doi.org/10.1039/TF9615701200

Mollerup, M., 2010. Daisy 2D Numerics.

Mollerup, M., Abrahamsen, P., Petersen, C.T., Hansen, S., 2014. Comparison of simulated water, nitrate, and bromide transport using a Hooghoudt-based and a dynamic drainage model. Water Resources Research 50, 1080–1094. https://doi.org/10.1002/2012WR013318

Nye, P.H., Tinker, P.B., 1977. Solute movement in the soil-root system, Studies in ecology. University of California Press, USA.

Wagenet, R.J., Hutson, J.L., 1989. LEACHM, A process-based model of water and solute movement, transformations, plant uptake and chemical reactions in the unsaturated zone (No. Version 2). Environmental research, Department of Agronomy, Cornell University, Ithace, New York.